UML 다이아그램간의 일관성과 완전성을 위한 검증 규칙 생성에 관한 연구

김재웅† · 김진수†† · 김치수††† · 황선명††††

요 약

객체지향 설계에 많이 사용되고 있는 UML의 일부 다이아그램들의 일관성과 완전성을 검증하기 위하여 UML 다이아그램들을 분석하여 ER 모델에 적용하고, 일련의 집합과 함수들을 사용하여 정형적으로 명세한 다음 이러한 함수들의 의미에 따라 UML 다이아그램들의 일관성과 완전성을 보장하는 일련의 규칙들을 유도한다. 이러한 규칙들은 추후에 다이아그램을 생성하는 능력과 일관성과 완전성을 검사하는 능력을 함께 가진 CASE 도구에 포함될 수 있다.

The Study on the Generation of Verifying Rules for Consistency and Completeness of the Unified Modeling Language Diagrams

Jae-Woong Kim†, Jin-Soo Kim††, Chi-Su Kim††† and Sun-Myung Hwang††††

ABSTRACT

For verifying consistency and completeness of some UML diagrams as a representative of the graphic notations for object-oriented designs, we first give an analysis of some UML diagrams and apply simple ER model to the UML diagrams. Next, we formally define this graphical representation as a set of sets and a set of functions. According to the semantics of these functions, we derive a set of rules to assure the consistency and completeness of this graphical representation. These rules will be included with a CASE tool, which possesses both the diagramming facility and the consistency and completeness checking capability.

1. 서 론

최근에 객체지향 방법이 소프트웨어를 개발하는 탁월한 파라다임이 되고 있고 객체지향 방법이 기존의 개발 방법들에 비해 개발되는 소프트웨어의 품질을 향상시키고 생산성을 증가시킨다는 것은 이미 많이 알려져 있다[1]. 소프트웨어 개발에서 나타나는 기본적인 문제점은 사용자 입장에서는 개발될 시스템이 어떨 것이라는 것을 정확하게 파악할 수 없고 개발자는 사용자의 요구를 완전하게 이해할 수 없다는 점이다. 결국 개발된 시스템들은 사용자가 요구한 기능들을 정확하게 만족시키기 어렵게 된다. 그러한 문제점을 해결하기 위해서는 언어로 구현되기 전에 사용자와 함께 개발자가 개발하고자 하는 시스템의 정확성을 확인하기 위해서 설계를 검토할 수 있어야 한다. 이러한 검토를 효과적으로 하기 위해서는 일반적으로 누구나 이해하기 용이한 다이아그램 형태로 설계를 표현하고 있다. 다이아그램들은 표현하기 쉽고 사용자에 의해 이해되기도 쉽고 설계자와 사용자간의 커뮤니케이션에도 편리하게 사용된다[2]. 객체지향 설계에서의 다이아그램의 일관성과 완전성은 전체적으로 논리적인 문제이며 컴퓨터에 의해 논리적인 추론을 수행하기 위해서는 일반적으로 해결해야만 하는 세가지 기본 문제가 있다. 첫째, 다이아그

† 정회원, 공주문화대학 컴퓨터정보과 교수
†† 한국전자통신연구원 초빙연구원
††† 정회원, 공주대학교 멀티미디어 연구소 부교수
†††† 대전대학교 컴퓨터공학과 부교수

램은 비정형화된 표현이기 때문에 다이아그램의 구문과 의미는 정형적으로 기술하기 어렵다. 둘째, 일반적으로 다이아그램은 컴퓨터에 의해 직접적으로 인식되고 처리될 수 없기 때문에 개발자는 이러한 다이아그램으로부터 필요한 정보를 유도해서 컴퓨터가 다이아그램 대신에 유도된 정보를 통하여 검사를 할 수 있어야 한다. 세째, 이러한 다이아그램들에 대해 어떻게 일관성과 완전성을 검사하는 가의 수행방법이다. 이러한 문제를 해결하기 위하여 본 논문에서는 OMG의 표준 객체지향 방법론으로 채택된 UML 다이아그램들에 대한 일관성과 완전성을 검사하기 위하여 각 다이아그램들을 ER 모델로 표현하고, 각 다이아그램을 하나의 그림으로 표현한 통합된 ER 다이아그램을 구성한다. 이렇게 구성된 통합 다이아그램은 집합과 함수들을 사용하여 정형적으로 명세되며, 이러한 정형 명세를 기반으로하여 일관성과 완전성 검사를 수행하기 위한 규칙을 제공한다. 이 규칙은 추후 CASE 도구에 포함되어 다이아그래밍의 일관성과 완전성을 검사할 수 있는 CASE 도구로 발전할 수 있게 된다.

2. 관련 연구

2.1 UML 다이아그램

객체지향 방법에서 사용되고 있는 다이아그램들은 사용하는 방법론 만큼이나 많다고 할 수 있다. 그러나 그려지는 모양은 다르게 보여지지만 표현하고 있는 관점이나 수준이 유사할 경우에는 동일한 개념에 대한 표현도 상당히 유사하다고 할 수 있다. 이러한 다이아그램의 유사성 때문에 어떠한 방법론의 다이아그램들도 본 논문에서 제시하고 있는 방법으로 적용이 가능할 것이다. 본 논문에서는 OMG의 표준 객체지향 방법론으로 채택된 UML(Unified Modeling Language)을 대상으로 사용되고 있는 일부 다이아그램들의 일관성 및 완전성을 검사하고자 한다. UML에서 사용되고 있는 다이아그램들은 문제를 바라보는 시각에 따라 다섯 그룹으로 나눌 수 있는데 [3,4] 본 논문에서는 이 가운데에 우선 클래스 다이아그램, 객체 다이아그램, 순차 다이아그램, 상태 다이아그램을 대상으로 다이아그램들간의 일관성과 완전성을 검사하는 규칙을 유도하고자 한다. 추후에 나머지 다이아그램들에 대한 일관성과 완전성도 포함할 예정이다.

2.2 ER 모델

ER 모델은 1976년에 Peter Chen[5]에 의해 처음 소개되었는데 초기의 이 모델은 엔티티, 관련성 및 속성을 이용하여 개념들을 표현하였다. 그 후 일반화 계층, 복합 속성 및 식별자 등의 다른 개념들이 이 모델에 추가되었다. ER 모델은 효과적인 표현력을 가지고 있기 때문에 정보시스템 설계 뿐만아니라 데이터베이스의 개념적 설계를 위한 산업계의 표준으로 선택되고 있다. 본 논문에서는 ER 모델을 사용하여 각 다이아그램들을 표현한다. ER 모델의 기본 요소는 엔티티 타입과 일반화 계층, 일대다 관련성이다. 엔티티 타입은 실세계의 유사한 엔티티들의 추상화라고 할 수 있다. 일반화 계층은 일반화 엔티티 타입과 상세화 엔티티 타입간의 일반화-상세화 관련성을 표현한다. 이 일반화-상세화 관련성은 화살표의 머리가 일반화 엔티티 타입으로 향하는 화살표로 표현한다. 실세계에서 엔티티들의 연관성은 엔티티 타입들간의 관련성으로 표현된다. 엔티티 타입의 한 엔티티가 다른 엔티티 타입의 하나나 그 이상의 엔티티와 연관이 된다면 이들 두 엔티티 타입간에는 일대다 관련성이 있다.

이러한 ER 모델이 실세계를 개념적으로 모델링하는데 편리한 도구이기 때문에 우리는 UML에 있는 일부 다이아그램들의 개념적 모델을 구축하기 위해서 ER 모델을 적용한다.

3. UML 다이아그램의 ER 모델

UML을 사용하여 시스템을 개발하는 경우, 개발되는 시스템은 여러 가지 다이아그램으로 표현될 수 있다. UML 다이아그램 그 자체가 잘 정의가 되어 있기는 하지만 이들 다이아그램들간에는 일관성과 완전성이 문제가 될 수 있다. 본 논문에서는 이러한 다이아그램들간의 일관성과 완전성을 검사하기 위하여 이들 다이아그램들을 고수준의 개념적 모델링을 사용하여 표현한 다음 정형 명세를 이용하여 검사하는 방법을 사용한다. 따라서 먼저 이들 각 다이아그램을 2.2절에서 설명한 ER 모델로 표현할 필요가 있다.

3.1 UML의 각 다이아그램에 대한 ER 모델

본 논문에서는 2.1절에 제시된 네 가지의 UML 다이아그램에 대한 ER 모델을 구축한 다음 각 다이아그램들간의 일관성과 완전성 검사를 위하여 이들 각각의 ER 모델을 하나의 ER 모델로 통합한다.

(1) 클래스 다이아그램에 대한 ER 다이아그램

클래스 다이아그램의 필수적인 요소는 클래스와 클래스들간의 관련성인데 클래스 엔티티 타입과 관련성 엔티티 타입을 사용하여 표현한다. 다음 그림 1은 클래스 다이아그램에 대한 ER 다이아그램을 보여주고 있다.
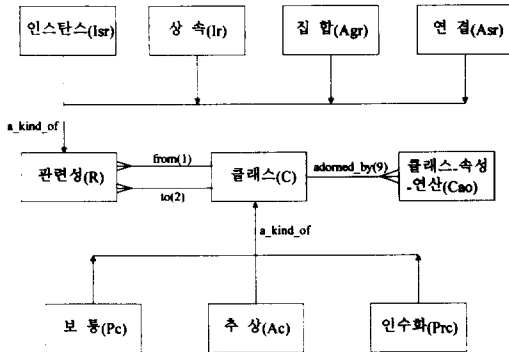
그림 1. 클래스 다이아그램에 대한 ER 다이아그램

(2) 객체 다이아그램에 대한 ER 다이아그램

객체 다이아그램의 필수적인 요소는 객체와 그들의 관련성이며 객체 엔티티 타입과 링크 엔티티 타입을 사용하여 표현된다. 다음 그림 2는 객체 다이아그램에 대한 ER 다이아그램을 보여주고 있다.
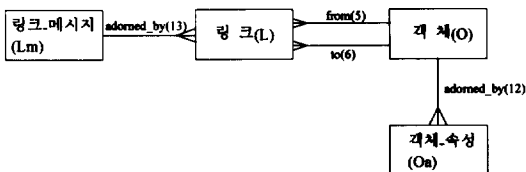
그림 2. 객체 다이아그램에 대한 ER 다이아그램

(3) 순차 다이아그램에 대한 ER 다이아그램

순차 다이아그램의 필수적인 요소는 객체와 그들간의 상호작용이며 객체 엔티티 타입과 상호작용 엔티티 타입을 사용하여 표현된다. 다음 그림 3은 순차 다이아그램에 대한 ER 다이아그램을 보여주고 있다.
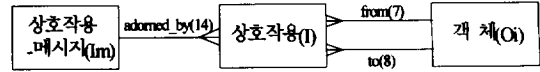
그림 3. 순차 다이아그램에 대한 ER 다이아그램

(4) 상태 다이아그램에 대한 ER 다이아그램

상태 다이아그램의 필수요소는 상태와 상태전이이며 상태 엔티티 타입과 상태전이 엔티티 타입으로 상태와 상태전이를 표현한다. 다음 그림 4는 상태 다이아그램에 대한 ER 다이아그램을 보여주고 있다.
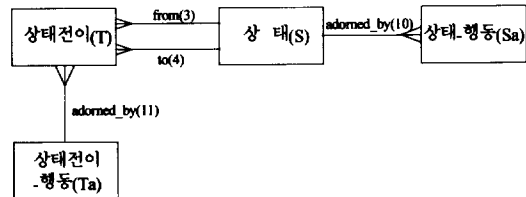
그림 4. 상태 다이아그램에 대한 ER 다이아그램

3.2 UML 다이아그램을 위한 통합된 ER 다이아그램

3.1에서 UML의 클래스 다이아그램, 객체 다이아그램, 순차 다이아그램, 상태 다이아그램에 대한 ER 다이아그램을 얻을 수 있었다. 하나의 ER 다이아그램에 있는 엔티티들은 다른 ER 다이아그램에 있는 엔티티들과 의미적으로 관련이 있다. 그러나 이러한 의미적 관련성은 각각의 ER 다이아그램에서는 나타나지 않는다. 여기에서는 각각의 ER 다이아그램에서 엔티티 타입들간의 의미적 관련성을 나타내기 위하여 분리된 각각의 다이아그램들을 하나의 ER 다이아그램으로 표현하고자 한다. 이들 각 ER 다이아그램에 있는 엔티티 타입들간의 의미적 관련성은 다음 그림 5와 같이 통합된 ER 다이아그램에서 엔티티들간의 여러 종류의 관련성으로 표현되고 있다. 통합된 ER 다이아그램은 UML에서 사용되고 있는 클래스 다이아그램, 객체 다이아그램, 순차 다이아그램, 상태 다이아그램을 위한 ER 다이아그램들의 조합으로 구축되었고 각 다이아그램 내부의 관련성은 표현하지 않았다.
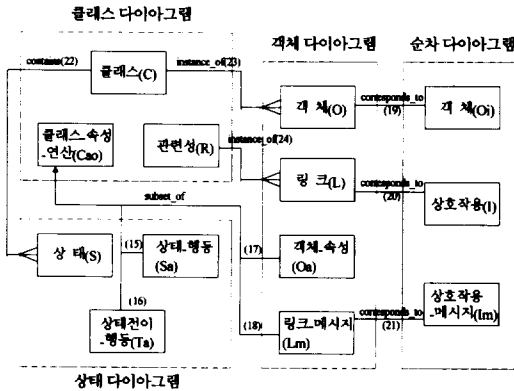
그림 5. 통합된 ER 다이아그램

4. UML 다이아그램의 일관성 및 완전성 검사 규칙

4.1 다이아그램의 정형 명세

3장에서 구축된 다이아그램에 대하여 집합과 함수를 이용하여 정형적으로 명세한 다음 함수들의 의미에 따라 다이아그램의 일관성과 완전성을 보장하기 위한 규칙들을 유도하고 이들 규칙에 정형 명세를 주고자 한다. 이들 다이아그램과 규칙들의 정형 명세는 수학적 기호로 표현된다. 그림 5.의 통합된 ER 다이아그램은 엔티티 타입과 이들 엔티티 타입들간의 관련성으로 구성된다. 따라서 이 다이아그램을 E와 R의 두 튜플로 정의할 수 있다.

UMLOOD = < E, R >
 E : 일반화된 엔티티 타입들의 집합
 R : 일반화된 엔티티 타입들간의 관련성의 집합

통합된 ER 다이아그램에서 각 일반화된 엔티티 타입은 집합 E의 한 요소이다. 따라서 첫 번째 투플 E는 다음과 같은 일반화된 엔티티 타입의 집합을 이용하여 정형적으로 정의된다.

$$E=\{C,R,Cao,S,T,Sa,Ta,O,L,Oa,Lm,I,Im\} \quad (1)$$

여기에서 제시된 집합 E의 요소인 엔티티 타입의 집합들은 추상자료형 집합을 사용하여 정형적으로 명세될 수 있다. UMLOOD의 정형화된 정의의 두번째 투플은 R이다. 투플 R은 다음과 같은 함수들로 정형적으로 정의된다.

$$R = \{RfC^{1}, RtC^{2}, TfS^{3}, TtS^{4}, LfO^{5}, LtO^{6}, IfOi^{7}, ItOi^{8}, CaoC^{9}, SaS^{10}, TTa^{11}, OaO^{12}, LLm^{13}, IIm^{14}, SaCao^{15}, TaCao^{16}, OaCao^{17}, LmCao^{18}, OiO^{19}, IL^{20}, ImLm^{21}, ScC^{22}, OcC^{23}, LcR^{24}\} \quad (2)$$

집합 R의 각 요소는 ER 다이아그램에서 엔티티 타입들간의 관련성을 설명하는 함수들로 정의된다.

4.2 다이아그램의 일관성과 완전성 규칙

본 논문에서의 일관성과 완전성은 객체지향 설계가 사용자의 요구를 만족시키기 위하여 제안된 시스템의 명세와 일관성이 있는지 없는지 또는 객체지향 설계가 실행가능한 시스템으로 구현되기 위하여 충분한 정보를 제공할 만큼 완전한지 아닌지를 의미하는 것은 아니다.

일관성과 완전성은 엄격하게 정의하기는 어렵지만 본논문에서의 일관성과 완전성은 UML 다이아그램의 모든 필수적인 요소들이 UML에서 사용되는 구문과 의미를 유지하고 있는지를 확인하는데 사용된다. 본 논문에서는 UML 다이아그램의 구문을 유지하는 일관성을 구문 일관성(syntactic consistency), UML 다이아그램의 의미를 유지하는 일관성을 의미 일관성(semantic consistency)이라고 한다. 또한 UML 다이아그램에 대한 완전성은 다이아그램의 모든 필수적인 요소들이 UML의 구문적이고 의미적인 요구사항들을 만족하는 완전한 방법에서 사용된다는 것을 의미한다. 본 논문에서는 UML 다이아그램의 구문적 요구사항을 만족하는 완전성을 구문 완전성(syntactic completeness), UML 다이아그램의 의미적 요구사항을 만족하는 완전성을 의미 완전성(semantic completeness)이라고 한다.

본 논문에서는 이러한 구문적 및 의미적 일관성과 완전성이 UML 다이아그램의 요소들 사이에 서로 연결된다는 것을 찾아내었다. 이들간의 연결은 3장에서 보여준 ER 다이아그램의 엔티티 타입들간의 관련성에 의해 모두 찾아지고 있다. 이들 관련성 각각은 4.1절에서 함수의 집합으로 명세된 하나의 함수로써 정형화되어 정의되고 있다. 따라서 이들 모든 집합들의 모든 요소가 이들 모든 함수들을 만족한다면, UML에서 객체지향 설계의 다이아그램들은 구문적으로나 의미적으로나 일관성이 있고 완전하다고 할 수 있다. 만약 이러한 함수중에 어떤 것을 만족하

지 않는 집합의 요소가 적어도 하나 이상 있다면, 이 다이아그램은 구문적 또는 의미적으로 일관성이 없거나 완전하지 않다고 할 수 있다. 따라서 본 논문에서는 이러한 다이아그램들의 일관성과 완전성의 정의를 다음과 같이 내리고 있다.

정의 : UML에서 객체지향 설계의 다이아그램이 구문적으로 의미적으로 일관성이 있고 완전하려면 4.1절의 (1)에 정의된 모든 집합의 모든 원소들이 4.1절의 (2)에 정의된 모든 함수들을 만족할 수 있어야 한다

이 정의에 따라 이 다이아그램의 일관성과 완전성을 검사한다는 것은 4.1절의 (1)에 정의된 이들 집합의 모든 원소가 4.1절의 (2)에 정의된 모든 함수들을 만족하는지를 보장하는 것이다. 만족하지 않는 어떠한 함수가 있다면, 이 다이아그램은 일관성이 없거나 또는 완전하지 않다라고 할 수 있다.

(1) 구문 완전성을 위한 규칙

객체지향 소프트웨어 시스템은 클래스들로 구성된다. 클래스들은 구조적인 방법 또는 의미적인 방법으로 서로 연결되어 있다. 이것은 클래스 다이아그램에서 모든 클래스들이 다양한 종류의 클래스 관련성에 의해 서로 연결되어 있다는 사실에 의해 알 수 있다. 어떠한 클래스도 다른 클래스들로부터 분리되어 있을 수는 없다. 클래스 다이아그램에서 모든 클래스들이 이러한 방법에 의해 그려졌다는 것을 보장하기 위하여, 다음과 같은 구문 완전성 규칙을 가진다.

규칙(11) 어떠한 클래스도 클래스 다이아그램에서 단독으로 존재할 수 없다.

이 규칙은 다음과 같이 정형적으로 명세될 수 있다.

$\forall c\ (\ c \in C \rightarrow \exists r\ (\ r \in R \wedge (\ RfC^{-1}(c) = r \vee RtC^{-1}(c) = r\ )))$

여기에서 C는 클래스들의 집합이고, R은 클래스 관련성의 집합을 나타내고, RfC와 RtC는 집합 R로부터 집합 C로의 함수를 나타낸다. 이와 유사하게 다음과 같은 규칙들을 생성할 수 있다.

규칙(12) 어떠한 객체도 객체 다이아그램에서 단독으로 존재할 수 없다

규칙(13) 어떠한 객체도 순차 다이아그램에서 단독으로 존재할 수 없다

규칙(14) 어떠한 상태도 상태 다이아그램에서 단독으로 존재할 수 없다

클래스 다이아그램에서 하나의 클래스 관련성은 클래스들간의 다양한 관련성을 나타내기 위하여 두 클래스들을 연결하는데 사용된다. 클래스 관련성의 한쪽 끝이 다른 클래스에 연결되지 않는다면 잘못된 것이다. 이것은 클래스 다이아그램에서 각 클래스 관련성은 두 클래스들을 연결해야 한다 라고 하는 UML 다이아그램의 구문적 요구사항에 의한 것이며 다음과 같은 구문적 완전성에 의해 식별될 수 있다.

규칙(111) 클래스 관련성의 양쪽 끝은 반드시 클래스들과 연결되어야 한다.

$\forall r\ (\ r \in R \rightarrow \exists ! c_i, c_j\ ((\ c_i \in C \wedge c_j \in C) \wedge (RfC(r) = c_i \wedge RtC(r) = c_j\ )))$

여기에서 R은 관련성들의 집합이고, C는 클래스의 집합을 나타내고, RfC와 RtC는 집합 R로부터 집합 C로의 함수를 나타낸다. 이와 유사하게 다음과 같은 규칙들을 생성할 수 있다.

규칙(112) 객체 링크 관련성의 양쪽 끝은 반드시 객체들과 연결되어야 한다

규칙(113) 객체 상호작용 관련성의 양쪽 끝은 반드시 객체들과 연결되어야 한다

규칙(114) 상태전이 관련성의 양쪽 끝은 반드시 상태들과 연결되어야 한다

(2) 구문 일관성을 위한 규칙

UML에서 보통 클래스는 인수화 클래스의 한 인스탄스이다. 이것은 클래스 다이아그램에서 단지 인수화 클래스와 보통 클래스를 연결하기 위해서 인스탄스 관련성을 사용할 때만 식별된다. 인스탄스 관련성, 인수화 클래스, 보통 클래스가 사용될 때 이러한 구문 제약을 표현하기 위해서 다음과 같은 두가지 구문 일관성 규칙을 생성한다.

규칙(21) 인스탄스 관련성의 꼬리는 인수화 클래스에만 연결되어 있어야 한다.

$\forall isr\ (\ isr \in Isr \rightarrow \exists ! prc\ (\ prc \in Prc \wedge RfC(isr) = prc\ ))$

여기에서 관련성의 부분 집합인 Isr은 인스탄스 관련성의 집합을 나타내고, 클래스 집합의 부분 집합인 Prc는 인수화 클래스의 집합을 나타낸

다. RfC는 집합 R로부터 집합 C로의 함수를 나타낸다.

규칙(22) 인스탄스 관련성의 머리는 보통 클래스에만 연결되어 있어야 한다

(3) 의미 완전성을 위한 규칙

클래스는 일종의 객체들의 모임이다. 만약 클래스가 객체 다이아그램에서 객체를 가지고 있지 않다면 이 클래스의 표현은 완전하지 않다. 이것은 각 클래스가 객체 다이아그램에서 하나의 객체를 포함해야 하는 UML 다이아그램의 의미 요구사항에 의한 것이며 다음과 같은 의미 완전성 규칙에 의해 식별될 수 있다.

규칙(31) 하나의 클래스는 객체 다이아그램에서 적어도 하나의 객체를 포함한다.

$\forall c\ (c \in C \rightarrow \exists o\ (o \in O \wedge OcC^{-1}(c) = o))$

여기에서 C는 클래스들의 집합을 나타내고, O는 객체들의 집합을 나타낸다. OcC는 집합 O로부터 집합 C로의 함수를 나타낸다.

(4) 의미 일관성을 위한 규칙

하나의 객체는 반드시 하나의 클래스에 포함된다. 이것은 하나의 객체가 하나의 클래스에 포함되어야 하는 UML 다이아그램의 의미 요구사항에 의한 것이며 다음과 같은 의미 일관성 규칙에 의해 식별될 수 있다.

규칙(41) 하나의 객체는 클래스의 한 인스탄스이다.

$\forall o\ (o \in O \rightarrow \exists !c\ (c \in C \wedge OcC(o) = c))$

여기에서 O는 객체들의 집합을 나타내고, C는 클래스들의 집합을 나타낸다. OcC는 집합 O로부터 집합 C로의 함수를 나타낸다. 이와 유사하게 다음과 같은 규칙들을 생성할 수 있다.

규칙(42) 하나의 상태는 클래스의 한 상태이다

규칙(43) 하나의 링크는 클래스 관련성의 한 인스탄스이다

클래스는 객체들의 모임이라고 할 수 있다. 이 객체들은 객체 다이아그램에 있는 일부의 객체들이다. 이것은 클래스의 한 객체는 객체 다이아그램에서의 한 객체이어야 한다는 UML 다이아그램의 의미 요구사항에 의한 것이며 다음과 같은 의미 일관성 규칙에 의해 식별될 수 있다.

규칙(411) 하나의 객체 속성은 클래스의 한 속성이다.

$\forall oa\ (oa \in Oa \rightarrow \exists !cao\ (cao \in Cao \wedge OaCao(oa) = cao))$

여기에서 Oa는 객체-속성의 집합을 나타내고, Cao는 클래스-속성-연산의 집합을 나타낸다. OaCao는 집합 Oa로부터 집합 Cao로의 함수를 나타낸다. 이와 유사하게 다음과 같은 규칙들을 생성할 수 있다.

규칙(412) 하나의 상태 행동은 클래스의 한 연산이다.

규칙(413) 하나의 상태전이행동은 클래스의 한 연산이다.

규칙(414) 하나의 링크 메시지는 클래스의 한 연산이다

규칙(415) 순차 다이아그램에서 한 객체는 객체 다이아그램에서의 한 객체이다

규칙(416) 순차 다이아그램에서 하나의 상호작용은 객체 다이아그램에서 하나의 링크이다

규칙(417) 순차 다이아그램에서 하나의 상호작용 메시지는 객체 다이아그램에서 하나의 링크 메시지이다

4.3 관련 연구와의 비교 평가

먼저 [6]에서는 객체지향 분석단계에서 사용자의 요구가 충분히 반영될 수 있도록 객체모델의 메소드와 상태전이 모델의 상태전이 사이의 상호참조를 통해 완전성과 일관성을 검증할 수 있는 도구를 개발하였다. 이 도구에서의 일관성과 완전성 검증은 객체모델의 메소드와 상태전이 모델의 상태전이를 상호참조하는 방식으로 국한되어 사용되고 있다.

[7]에서는 지식베이스를 이용하여 객체지향 분석모델들에 대한 오류와 일관성 검증 방법을 제안하고 있다. 제안한 방법은 모형화 단계, 정형화 단계, 검증단계의 세 단계로 이루어져 있으며 모형화 단계에서 시스템을 분석하여 OMT 방법론의 세가지 모델을 생성하고 정형화 단계에서 이 세가지 모델들을 Atomic Formula 형태로 정형 명세하여 응용 지식베이스에 저장하고 마지막으로 검증 단계에서 오류 검출 규칙과 일관성 점검 규칙을 이용하여 오류를 점검하고 일관성을 유지한다. 이 방법은 각 모델들에 대한 정형 명세와 규칙은 생성하였지만 모델들간의 일

관성과 완전성에 대해서는 고려하고 있지 않다.

[8]에서는 UML 다이아그램들에 대한 메타 모델을 작성하고, 메타 모델의 각 요소들에 대하여 적용할 일반화된 메타 규칙을 도출하고, 메타 규칙들을 각 다이아그램에 적용한 세부 규칙 도출에 활용하였다. 그러나 이 방법에서는 각 규칙에 대한 정형화 작업 대신에 Prolog언어를 활용하여 규칙기반 시스템으로 설계모델을 검사하고 있다.

[9]에서는 UML 정적구조 다이아그램으로부터 LOTOS 명세를 생성하는 방법을 제시하고 있으나 정적 모델에 국한되어 LOTOS 명세를 생성하였고, 모델간의 일관성과 완전성에 관한 규칙은 제시하지 않았다.

[10]에서는 객체지향 모형간 일관성 검증을 지원하는 CASE 도구를 설계하고 구현하였으나 구체적인 검증 규칙에 대한 정형 명세가 제공되지 않는다.

본 논문에서는 UML 다이아그램들간의 일관성 및 완전성 검사를 위한 규칙들을 체계적으로 유도하였다. 이러한 규칙들을 유도하기 위하여 UML의 각 다이아그램들을 ER 모델에 적용하였고, 각 다이아그램에 대한 통합된 ER 다이아그램을 대하여 집합과 함수들로 정형적으로 정의하였으며 이 함수들의 의미에 기반하여 다이아그램들의 구문과 의미에 따라 일관성과 완전성을 보장하기 위한 일련의 규칙들을 유도하였다.

5. 결 론

객체지향 방법에서 시스템에 대한 설계의 결과를 시각화하기 위하여 사용하고 있는 많은 다이아그램들은 시스템 개발자 뿐만아니라 사용자가 충분히 이해하기 위해서 다양한 측면을 표현할 수 있어야 한다. 그러나 이러한 다이아그램들이 다양한 측면으로 표현되기 때문에 다이아그램들간에는 일관성과 완전성에 대한 문제가 발생할 수 있다. 본 논문에서는 이러한 다이아그램간의 일관성과 완전성을 유지할 수 있도록 일관성과 완전성을 검사할 수 있는 규칙을 생성하는 것을 목적으로 하고 있다.

본 논문에서는 UML의 각 다이아그램들의 요소들을 분석하고, UML 다이아그램들을 ER 모델에 적용하였다. 또한 개념적 모델링을 기반으로 일련의 집합과 함수들을 사용하여 정형적으로 명세한 다음 함수들의 의미에 따라 UML 다이아그램들의 일관성과 완전성을 보장하는 일련의 규칙들을 유도하였다. 이렇게 유도된 규칙들은 정형명세 언어로 쉽게 변환될 수 있을 뿐만아니라 변환된 정형명세 언어를 이용하여 다이아그램의 일관성과 완전성을 검사할 수 있는 검증 도구에 활용될 수 있다.

향후 연구과제로는 본 논문에서 유도된 규칙들을 이용하여 자동으로 다이아그램들의 일관성과 완전성에 대한 검증이 가능한 CASE 도구의 생성과 UML 다이아그램 뿐만아니라 다른 방법론의 다이아그램에 대한 일관성과 완전성 검증에 적용이 가능한 도구의 개발이다.

참 고 문 헌

[1] 김수동, 실무자를 위한 소프트웨어 공학, 에드텍, 1999

[2] Edwards, J. M. and Henderson-Sellors, B., "A graphical notation for object-oriented analysis and design", *Journal of Object-Oriented Programming,* Vol 4. No. 9 , pp.53-74, 1994

[3] Paul Hramon and Mark Watson, *Understanding UML : The Developer's Guide*, Morgan Kaufmann Publishers, Inc., 1997

[4] Martin Fowler, *UML Distilled*, Addison Wesley, 1997

[5] Chen, P. P., The entity-relationship model: towards a unified view of data, *ACM Transactions on Database System,* 1, pp. 9-36, 1976

[6] 김치수, 진영진, "객체지향 분석의 완전성과 일관성 검증을 위한 툴의 설계", 한국정보처리학회 논문지, 제4권, 제10호, pp. 2453-2460, 1997

[7] 김도형, 정기원, "객체지향 분석과정에서 오류와 일관성 점검방법", 정보과학회논문지(B), 제26권, 제3호, pp. 380-391, 1999

[8] 정기원, 조용선, 권성구, "객체지향 설계방법에서 오류 검출과 일관성 점검기법", 한국정보처리학회 논문지, 제6권, 제8호, pp. 2072-2087, 1999

[9] 김철홍, 안유환, 이원천, "UML 정적구조 다이아그램으로부터 LOTOS 명세 생성", 한국정보

처리학회 논문지, 제6권, 제12호, pp. 3500-3513, 1999

[10] 이선미, 전진옥, 류재철, "객체지향 모형간 일관성 검증을 지원하는 CASE 도구 설계 및 구현", 한국정보처리학회 논문지, 제6권, 제11호, pp. 2965-2980, 1999

김 재 웅

1983년 중앙대학교 공과대학 전자계산학과 (학사)
1988년 중앙대학교 공과대학 전자계산학과 (석사)
2000년 8월 대전대학교 컴퓨터공학과 (박사 학위 예정)
1992년 ~ 현재 공주문화대학 컴퓨터정보과 부교수
관심분야 : 소프트웨어 공학, 객체지향 방법론, 컴포넌트 기반 소프트웨어 개발방법론, 원격교육 시스템

김 진 수

1986년 중앙대학교 공과대학 전자계산학과 (학사)
1988년 중앙대학교 공과대학 전자계산학과 (석사)
1997년 중앙대학교 공과대학 컴퓨터공학과 (박사)
1998년 ~ 현재 건양대학교 정보전자통신공학부 조교수
1998년 ~ 현재 한국전자통신연구원 초빙연구원
관심분야 : 소프트웨어 공학, 객체지향 방법론, 소프트웨어 품질보증, 컴포넌트 기반 소프트웨어 개발방법론, 원격교육 시스템

김 치 수

1984년 중앙대학교 전자계산학과 (학사)
1986년 중앙대학교 전자계산학과 (석사)
1990년 중앙대학교 전자계산학과 (박사)
1990년 9월 ~ 1992년 8월 공주교육대학교 전임강사
1992년 ~ 현재 공주대학교 멀티미디어 연구소 부교수
관심분야 : 객체지향 방법론, 컴포넌트 개발 방법론

황 선 명

1987년 중앙대학교 전자계산학과 박사
1988년 독일 Bonn대학 Post Doctor
1989년 ~ 현재 대전대학교 컴퓨터공학과 부교수
KSPICE 운영위원 및 심사원, TTA S/W 개발기술연구원
관심분야 : 소프트웨어 품질보증, 테스팅 방법 및 도구, 표준화, 재공학